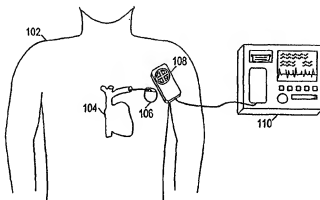




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : A61N 1/372	A1	(11) International Publication Number: WO 00/29064 (43) International Publication Date: 25 May 2000 (25.05.00)
(21) International Application Number: PCT/US99/27056 (22) International Filing Date: 12 November 1999 (12.11.99) (30) Priority Data: 09/191,808 13 November 1998 (13.11.98) US (71) Applicant: INTERMEDICS INC. [US/US]; 4000 Technology Drive, Angleton, TX 77515 (US). (72) Inventors: FOX, James, K.; 305 Timbercreek Drive, Lake Jackson, TX 77566 (US). ROTTENBERG, William, B.; 106 Paintbrush, Lake Jackson, TX 77566 (US). THOME, Bryan, J.; Apartment 307, 301 Cannon, Angleton, TX 77515 (US). (74) Agent: VIKSNINS, Ann, S.; Schwegman, Lundberg, Woessner & Kluth, P.O. Box 2938, Minneapolis, MN 55402 (US).		(81) Designated States: AU, CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: IMPLANTABLE DEVICE AND PROGRAMMER SYSTEM WHICH PERMITS MULTIPLE PROGRAMMERS



(57) Abstract

A multiprogrammer system for monitoring and optimizing implantable device performance. In one embodiment, the system includes at least two programming devices and an implantable device. Each of the programming devices may be used to perform inquiry and programming operations on the implantable device. In an inquiry operation, the programming device retrieves some or all of the configuration parameters from the implantable device. In a programming operation, the programming device provides one or more modified parameters to the implantable device. As part of the programming operation, the programming device is configured to verify that it is aware of the implantable device's current parameters before sending the modified parameters. In other words, the current programming device verifies that the implantable device's parameters have not been altered by another programming device since the current programming device's last interaction with the implantable device. If the parameters have been altered, the current programming device aborts the programming operation and notifies the operator. In alternate embodiments, the verification may be performed by the implantable device, i.e. the implantable device may verify that the programming device is aware of the current device parameters before the implantable device accepts the modified parameters.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroun			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

IMPLANTABLE DEVICE AND PROGRAMMER SYSTEM WHICH PERMITS MULTIPLE PROGRAMMERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to programming of implantable devices, and more particularly to a programming system which addresses the dangers raised by the presence of multiple programming devices.

2. Description of the Related Art

Implantable devices have become a standard method of treating various medical conditions, many of which relate to the heart. Examples of implantable devices include pacemakers, defibrillators, nerve stimulators, drug delivery devices, and implanted personal identification chips. Many types of implantable devices are available with high capacity memories for storing data and various programmable configuration parameters. In the case of medical devices, the data to be stored may include physiological data such as the electrogram (electrical waveform of the heart detected at the electrodes), instantaneous heart rate, blood pressure, volume pumped, body temperature, etc. Configuration parameters that are stored may include modes of operation, amplifier sensitivity, filter bandwidth, adaptation algorithms, output voltages, currents and pulse widths, blanking periods, various pacing rates, circadian response patterns, lead characteristics, delay intervals, detection thresholds, safety margins, logging criteria, and error messages. As implantable devices increase in sophistication, the number of configuration parameters is also expected to increase.

Referring now to Figure 1, a human torso 102 is shown having an implantable device 106 coupled to a heart 104. When a wand 108 from an external programming device 110 is placed in proximity to implantable device 106, the programming device 110 can establish two-way communication with implantable device 106 to retrieve data and to provide new configuration parameters. Often the device 106 collects data over a period of hours or days. In the case of a pacemaker, the data may represent measured physiological signals such as cardiac voltages (EKG signals), blood temperatures, oxygen levels, sugar levels, and other physical parameters.

Illustratively, the programming device 110 comprises an implantable device programmer and data analyzer that is used by a physician. The programmer/analyzer operates to download information stored in implantable device 106 by transmitting signals which place the pacer in a mode for downloading, and thereafter detecting signals sent by the device. Then, under control of the physician or other medical professional, the programmer/analyzer operates to analyze and display the information in a format which allows the physician to diagnose any problems. After performing an analysis, the physician may instruct the programmer/analyzer to adjust operating parameters for a different mode of operation, sensitivity setting, or other parameter value, to tailor the behavior of the device to the patient and thereby optimize the patient's quality of life. If this is the case, the programmer/analyzer 110 provides new operating parameters to the implantable device 106.

Implant manufacturers have long been aware of a danger known as the "multiple programmer" problem which can result in an implantable device having incorrect and perhaps even dangerous or harmful configuration parameters. The following scenario is presented to illustrate this problem.

A patient with an implantable device enters an examination room, and as part of a routine initial examination has a medical technician use a first programming device "A" to download data and configuration parameters from the implantable device. Depending on the amount of data and the baud rate of the device, the download time may range from 20 seconds to 20 minutes. In the absence of any gross abnormalities in the downloaded data, the patient is sent to another room for an exercise session to determine "rate modulated" settings, *i.e.* configuration parameters for adjusting the pacemaker pace rate in response to detecting patient exertion. In this room, a physician uses a second programming device "B" to download and adjust the configuration parameters in response to the results of the exercise session. Programming device "B" is used to reprogram the implantable device with the adjusted parameters. The patient then returns to the examination room, where a physician uses programming device "A" to adjust some of the configuration parameters in response to analysis of the downloaded data. Programming device "A" is then used to reprogram the implantable device with the adjusted parameters. It is important to note that programming device "A" is, at this point,

operating with an obsolete version of the implant's configuration settings. This situation occurs whenever changes are made to the implantable device's configuration parameters by a second programming device "B" between the download and reprogramming operations of the first programming device "A".

5 The configuration parameters of an implantable device such as a pacemaker can individually be set to typical values within a normal operating range, but the programming device must still check for incompatible parameter settings to avoid dangerous combinations of parameter values. For example, programming an inappropriately long refractory period in conjunction with a short pacing cycle may lead
10 to unpredictable pacing behavior. If a programming device with an obsolete version of the implant device's configuration settings reprograms only a few parameters, any safeguards that the programming device implements to avoid incompatible parameter settings could be unintentionally circumvented.

 One programmer safeguard that has been employed is to have programming
15 device "A" reprogram the implantable device with a complete set of configuration parameters rather than just the parameters which have been adjusted. Although this successfully prevents incompatible configuration settings, the previous adjustments are completely undone without any indication to the physician. Further, the reprogramming time is unnecessarily increased beyond what may be strictly necessary.

20 The solution commonly employed by implant manufacturers has been simply to issue warnings regarding the danger of using multiple programming devices. A more effective and inexpensive solution to the multiple programming device problem is desirable.

25

SUMMARY OF THE INVENTION

 Accordingly, there is provided herein a multiprogrammer system for monitoring and optimizing implantable device performance. In one embodiment, the system includes at least two programming devices and an implantable device. Each of the programming devices may be used to perform inquiry and programming operations on the implantable
30 device. In an inquiry operation, the programming device retrieves some or all of the configuration parameters from the implantable device. In a programming operation, the

programming device provides one or more modified parameters to the implantable device. As part of the programming operation, the programming device is configured to verify that it is aware of the implantable device's current parameters before sending the modified parameters. In other words, the current programming device verifies that the
5 implantable device's parameters have not been altered by another programming device since the current programming device's last interaction with the implantable device. If the parameters have been altered, the current programming device aborts the programming operation and notifies the operator. In alternate embodiments, the verification may be performed by the implantable device, *i.e.* the implantable device may verify that the
10 programming device is aware of the current device parameters before the implantable device accepts the modified parameters.

As part of the multiprogrammer system provided herein, there is disclosed a programming device embodiment, an implantable device embodiment, and various methods for verifying that the programming device has been provided with a current
15 version of the implantable device's configuration parameters. The programming device preferably comprises a user interface, a memory, a communications circuit, and a microprocessor. The communications circuit generally includes a communication coil, receive sensor, modulator, and demodulator, and is configurable to send and receive configuration parameters to and from the implantable device. The microprocessor couples
20 to the user interface, the memory, and the communications circuit, and it operates under control of the user interface to retrieve configuration parameters from the implantable device and to store the parameters in memory. The configuration parameters may be modified, and the modified parameters can be sent to the implantable device. In certain disclosed embodiments, the microprocessor uses one of the following methods to verify
25 that the stored configuration parameters are "current" before sending the modified parameters, that is, the microprocessor verifies that the implantable device's parameters have not been altered since this programmable device last retrieved the implantable device's parameters.

The implantable device comprises a microprocessor coupled to a memory and a
30 telemetry module. The memory stores configuration parameters, and the telemetry module transmits and receives external communications. The microprocessor performs

some algorithm in a manner governed by the configuration parameters stored in the memory. In certain disclosed embodiments, the microprocessor uses one of the following methods to verify that an external programming device has received a current version of the configuration parameters before accepting modified configuration parameters from the programming device.

The disclosed methods for verifying that a programming device is aware of the implantable device's current configuration parameters include: (1) providing a programming device serial number to the implantable device as part of every inquiry operation; (2) allowing a programming operation only within a predetermined time interval after an inquiry operation; (3) storing the date and time of the most recent inquiry operation; (4) storing the date and time of the most recent programming operation; and (5) retrieving the implantable device's current parameters as part of every programming operation. For method (1), the implantable device or the programming device can be configured to compare the programming device's serial number with the serial number of the last programming device to retrieve the implantable device's configuration parameters. A match indicates that the programming device is aware of the implantable device's current parameter values.

For method (2), the implantable device or the programming device can determine if the downloaded version of the configuration parameters has "expired", that is, whether a predetermined amount of time has passed since the configuration parameters were last retrieved. For method (3), the implantable device or programming device can be configured to compare the date and time of the most recent inquiry stored in the programming device to that stored in the implantable device. A match indicates that the programming device is aware of the implantable device's current parameter values.

Similarly, for method (4), the implantable device or programming device can be configured to compare the date and time of the most recent programming operation stored in the programming device to that stored in the implantable device. Finally, for method (5), the programming device can be configured to retrieve the implantable device's configuration parameters immediately prior to a programming operation to verify that the programming device has a current version. These methods will be described in more detail further below.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

- 5 Figure 1 shows an implantable medical device and a programming device;
 Figure 2 is a block diagram of an implantable pacemaker;
 Figure 3 is a block diagram of an exemplary embodiment of a programming device;
 Figure 4 is a flowchart depicting a method for performing inquiry and
10 programming operations;
 Figure 5 is a flowchart illustrating the multiprogrammer problem;
 Figure 6 is a flowchart depicting a first method for performing inquiry and programming operations in a multiprogrammer environment;
 Figure 7 is a flowchart depicting a second method for performing inquiry and
15 programming operations in a multiprogrammer environment;
 Figure 8 is a flowchart depicting a third method for performing inquiry and programming operations in a multiprogrammer environment;
 Figure 9 is a flowchart depicting a fourth method for performing inquiry and programming operations in a multiprogrammer environment;
20 Figure 10 is a flowchart depicting a fifth method for performing inquiry and programming operations in a multiprogrammer environment;
 Figure 11 is a flowchart depicting a sixth method for performing inquiry and programming operations in a multiprogrammer environment;
 Figure 12 is a flowchart depicting a seventh method for performing inquiry
25 and programming operations in a multiprogrammer environment;
 Figure 13 is a flowchart depicting an eighth method for performing inquiry and programming operations in a multiprogrammer environment; and
 Figure 14 is a flowchart depicting a ninth method for performing inquiry and programming operations in a multiprogrammer environment.
30 While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of examples in the drawings and will

herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description illustrates the principles of the present invention with respect to an implantable pacemaker ("pacer") and a programming device ("programmer"). The invention, however, is directed to a system for permitting multiple programming devices to interact with an implantable device. Thus, the invention applies to implantable cardioverter/defibrillators (ICD's), nerve stimulators, drug delivery devices, or any other implantable device which may have programmable configuration parameters set by a programming device.

Referring now to Figure 2, an exemplary implantable device 106 (such as a pacemaker) preferably includes a power supply 202 coupled to a microprocessor 204. The power supply 202 provides power to all the devices shown in Figure 2 through connections which are not specifically shown. In the exemplary embodiment, the microprocessor 204 couples to a memory 206, a first interval timer 208, and a second interval timer 210 via an I/O (input/output) bus 211. The microprocessor 204 also couples to control an atrium sensor/stimulator 212 and a ventricle sensor/stimulator 214, each of which may be coupled to the heart by flexible leads. Finally, microprocessor 204 couples to a telemetry module 218 to communicate with programming device 110.

The microprocessor 204 preferably is programmable and operates according to a program stored preferably in a nonvolatile memory such as a read-only memory (not specifically shown). The program is parameterized - *i.e.* one or more of the operations the microprocessor performs is alterable by setting a configuration parameter. For example, the microprocessor may be programmed to periodically trigger the atrium sensor/stimulator 212 to deliver a pulse to the heart 104. In this instance, one of the configuration parameters for this operation is the maximum trigger delay, that is, a value specifying the maximum time delay before the atrium stimulator is triggered. The

maximum trigger delay, along with other configuration parameters, is provided to microprocessor 204 via telemetry module 218 and stored in memory 206. During pacemaker operation, the microprocessor 204 retrieves the maximum trigger delay from the memory 206 and uses it to set interval timer 210 after each triggering of the atrium sensor/stimulator. If the interval timer 208 expires, the microprocessor then triggers atrium sensor/stimulator 212.

Referring still to Figure 2, the first interval timer 208 determines the delay between trigger signals applied to atrium stimulator 212 and ventricle stimulator 214. The second interval timer 210 measures the time since the last heartbeat sensed by the atrium sensor/stimulator 212 or ventricle sensor/stimulator 214. When either timer elapses, the elapsed timer asserts an interrupt signal to microprocessor 204 to notify the microprocessor 204 that the set amount of time has passed. Microprocessor 204 determines the source of the interrupt according to conventional techniques and takes the appropriate action. For example, if the maximum trigger delay (interval timer 210) has elapsed since the last heartbeat, the microprocessor 204 triggers atrium sensor/stimulator 212.

The microprocessor 204 also preferably monitors one or more physiological signals. In the pacemaker example, the microprocessor 204 detects cardiac voltage signals via atrium sensor 212 and/or ventricle sensor 214. The heart leads which provide electrical pulses to the heart may also be used to sense electrical signals created by the heart as it beats, and these signals are used by the microprocessor 204 to adjust the timing of the electrical pulses. When other sensors are coupled to the implantable device 106, the microprocessor can also monitor blood pressure, body temperature, oxygen levels, and other physiological parameters. The microprocessor 204 can also track its own performance, perhaps by logging the manner in which adaptation is performed on the parameters. The physiological signals and performance information can be logged in memory 206 for later retrieval by programming device 110. The memory 206 preferably is large enough to store data regarding several physiological signals that being are monitored over a period of several days. Memory 206 preferably is implemented as dynamic random access memory (DRAM) or other suitable memory type.

The atrium sensor/stimulator 212 is an interface circuit between microprocessor 204 and a heart lead coupled to an atrium of the heart. An interface circuit is necessary to allow the digital, low voltage microprocessor to control the high-energy pulses delivered to the heart, and additionally, to allow the microprocessor to monitor the analog electrical signals that are generated by the heart as it beats. Similarly, the ventricle sensor/stimulator 214 is an interface circuit between microprocessor 204 and a heart lead that couples to a ventricle of the heart. When atrium sensor/stimulator 212 receives a trigger signal from microprocessor 204, it generates a shaped electrical energy pulse to the atrium. Likewise, when ventricle sensor/stimulator 212 receives a trigger signal from microprocessor 204, it provides a shaped electrical energy pulse to the ventricle. If the microprocessor is using atrium sensor/stimulator 212 or ventricle sensor/stimulator 214 to measure cardiac voltage signals from the electrodes to monitor the performance of the heart, the microprocessor 204 stores the cardiac waveforms (or "electrograms") in memory for subsequent retrieval by a medical technician.

Telemetry module 218 may be designed to be activated by programming device 110 when wand 108 enters into proximity with pacer 106. For example, the telemetry module 218 may continually be checking for an activation signal that the wand 108 transmits. Activation of the telemetry module 218 causes the telemetry module 218 to establish bi-directional communication with wand 108 and to notify microprocessor 204 of an incoming communication. As the wand 108 transmits a message signal, the telemetry module demodulates the message signal and delivers the incoming message to the microprocessor 204. The microprocessor 204 decodes the incoming message and stores any received data or parameters. In addition, the microprocessor 204 responds to any received commands from the programming device 110. For example, one command might be an "inquiry", that is, a request for the microprocessor 204 to transfer configuration parameters from memory 206 to programming device 110. In this case, microprocessor 204 provides the configuration parameters from memory 206 to telemetry module 218 for transferal to programming device 110.

Referring now to Figure 3, programming device 110 includes a microprocessor 302, a modulator 304 coupled to a transmit coil 306, a demodulator 310 coupled to a receive sensor 308, a memory 312, and a user interface 314. The microprocessor 302

responds to user input via the user interface 314 (which may comprise a graphic display and user input device such as a keypad) and initiates communications with pacer 106 (Fig. 2). For example, if a user requests a download of data from the pacer to programming device 110, microprocessor 302 formulates a command signal, and sends the signal to modulator 304. Modulator 304 converts the command signal into a modulated signal for driving transmit coil 306. The signal driving the transmit coil 306 produces a changing magnetic field which induces a current in a receive coil in the pacer. The pacer processes the induced current to reconstruct the information sent from the programming device, and formulates and sends a reply. The pacer can transmit signals to programming device 110 by various means including modulating a light signal or driving a transmit coil. Receive sensor 308 detects and amplifies the signal transmitted by the pacer to produce a detection signal. Demodulator 310 demodulates the detection signal and converts it into the data transmitted by the pacer 106. Demodulator 310 then provides the data to microprocessor 302 for eventual analysis and display via user interface 314. Memory 312 may be used to store data and configuration parameters downloaded from the pacer.

Referring now to Figure 4, a pair of flowcharts are provided to illustrate the operation of the processor 302 in the programming device 110 and the operation of the processor in 204 the implantable device 106 during the inquiry and programming operations. The programming device's processor 302 starts in step 401 and the implantable device's processor 204 starts in step 411. In step 402 the processor 302 initiates an inquiry operation by sending a command to the implantable device requesting the implantable device to transmit its model and serial number. In step 403 the processor 302 receives the model and serial number information and stores it for future identification of the implantable device. In step 404 the processor 302 sends a request to the implantable device for the implantable device to transmit one or more of its configuration parameters. The parameter request may be a partial request, in which only certain specified parameters are requested, or a full request, in which a download of all the configuration parameters is requested. In a typical initial inquiry operation, the parameter request is a full request. In step 405 the processor 302 receives the transmitted configuration parameters and stores them in memory for analysis and possible alteration

by the user of the programming device 110. This may complete the inquiry operation, or as a further part of the inquiry operation the processor 302 may request and receive stored physiological data from the implantable device.

After a user examines the configuration parameters and analyzes any downloaded data, the user may wish to modify one or more of the configuration parameters and to perform a programming operation to reprogram the implantable device with the modified parameters. In step 406 the processor 302 initiates a programming operation by sending a command to the implantable device requesting the implantable device to transmit its model and serial number. In step 407 the processor 302 receives the model and serial number information and verifies that it matches with the stored identification information from step 403. The processor then in step 408 transmits the modified parameters. The programming device transmits a partial or complete parameter set to the implantable device. A partial parameter set is preferred for a faster programming operation, but a complete parameter set is preferred for added safety against incompatible configuration parameter settings. In step 409 the processor 302 receives confirmation from the implantable device that the transmitted parameters were successfully received. If confirmation is not received, in step 409, then the user is notified of a failure to reprogram the implantable device. The processor 302 completes the programming operation by entering into end state 410.

In step 412 processor 204 of the implantable device receives the model and serial number request command sent by the programming device in step 402. In step 413 the processor 204 responds by transmitting the model and serial number of the implantable device. In step 414 the processor 204 receives the configuration parameter request command from the programming device, and in step 415 the processor responds by transmitting the current configuration parameter values. Subsequently during a programming operation, the processor 204 receives another model and serial number request command in step 416, and responds in step 417 by transmitting the model and serial number of the implantable device. Then in step 418 the processor 204 receives modified parameter values from the programming device. After verifying that the parameters have been correctly received, the processor updates the configuration

parameters with the new values and, in step 419, transmits a confirmation message to the programming device.

Figure 5 illustrates how the use of multiple programming devices can introduce safety concerns when a second programming device interacts with an implantable device between the inquiry and programming operations of the first programming device. An implantable device begins in step 502. In step 504, an inquiry operation is performed by a first programming device "A". Subsequently, a second programming device "B" performs an inquiry operation on the implantable device in step 506, and a programming operation on the implantable device in step 508. The first programming device "A" is then used to program the implantable device in step 510, and at the end of this sequence in step 512, the implantable device may be left with an undesired set of configuration parameters. To avoid this, programming device "A" or the implantable device may be provided with a means for detecting if another programming device interacted with the implantable device between the inquiry and program operations of programming device "A".

A first embodiment of a multiple programming device-implantable device system is shown in Figure 6. The programming device's processor 302 proceeds through the inquiry operation steps 401-405 as outlined previously. After the processor 302 successfully receives all the requested parameters in step 405, in step 601 the processor 302 sends a programming device identification number to the implantable device for the implantable device to store. The programming device identification number preferably includes the model number and serial number of the programming device. In this system embodiment, the implantable device always has a record of the last programming device to successfully perform an inquiry or programming operation on it.

Subsequently, when the programming device is ready to transmit altered parameters to the implantable device, the processor 302 proceeds through steps 406, 407, and after verifying that the implant identification information matches, the processor transmits the programming device identification number in step 602 before transmitting the modified configuration parameters in step 408 and proceeding through steps 409 and 410.

The implantable device's processor 204 similarly proceeds through inquiry operation steps 411-415 as outlined previously. After the processor 204 transmits the requested parameters in step 415, it receives and stores the programming device identification information in step 603. Then when the programming device initiates a programming operation, the processor 204 performs steps 416, 417 before receiving the programming device's identification information in step 604. In step 606 the processor 204 compares the received identification information to the previously stored identification information from step 603. If they do not match, then in step 608 the processor 204 transmits a denial to the programming device and ends the programming operation in step 420. If the information matches, then the processor 204 performs steps 418-420 to accept and confirm the modified parameters.

This system embodiment provides that only the programming device which has most recently interacted successfully with the implantable device is allowed to program the implantable device. Referring back to Figure 5, in step 510 this embodiment of programming device "A" would receive a denial from the implantable device when a programming operation is attempted. Programming device "A" would preferably inform the user of the denial and allow the user to initiate an inquiry operation to discover any alterations which may have been made to the configuration parameters.

Another system embodiment is illustrated in Figure 7. The inquiry operation of this embodiment matches that of the previous embodiment for both the programming device and the implantable device. After step 417 of the programming operation, the implantable device's processor 204 sends the programmer identification information to the programmer device in step 705. Then in step 418, the processor 204 receives any transmitted parameters and proceeds through steps 419, 420 as outlined previously.

After step 407 of the programming operation the programming device's processor 302 receives in step 701 the identification information of the last programmer to interact with the implantable device. In step 703 the processor 302 verifies that the identification information matches the programming device's model and serial number. If they do not match, the processor 302 terminates the programming operation in step 410 and preferably notifies the user of the error. Otherwise, the processor proceeds through steps 408-410 of the programming operation to provide the modified parameters to the

implantable device. This embodiment advantageously places fewer demands on the implantable device, thereby minimizing implementation cost.

Another programming device embodiment is shown in Figure 8. In this embodiment, after successfully receiving the configuration parameters in step 405, the programming device's processor 302 starts a timer in step 801. Subsequently, before initiating a transmission of altered parameters, the processor 302 checks for expiration of the timer in step 803. If too much time has elapsed, the processor 302 aborts the programming operation and preferably notifies the operator. Otherwise, the processor 302 continues with the programming operation in steps 406-410. An implantable device embodiment which operates similarly is shown in Figure 9. After successfully completing transmission of requested parameters, the implantable device's processor 204 starts a timer in step 902. After a programming operation is initiated, the processor 204 checks for expiration of the timer in step 904. If too much time has elapsed, the processor 204 transmits a denial in step 906, and the programming device preferably notifies the operator of the failure of the programming operation. Otherwise, the processor 204 continues the programming operation in steps 418-420. In a variation on the embodiments of Figs. 8 and 9, the timers may be restarted after successful completion of the programming operation. The timers in Figs. 8 and 9 preferably run for a time period greater than 5 minutes and less than 60 minutes. A time period of between 10 and 20 minutes is contemplated.

Two more system embodiments are shown in Figs. 10 and 11. These embodiments are closely related to those of Figs. 6 and 7, respectively. However, rather than using the programming device's identification number to identify the programming devices, these embodiments use the time and date of last access to identify the programming devices.

Referring now to Fig. 10, after the programming device's processor 302 successfully receives all the requested parameters in step 405, in step 1001 the processor 302 stores and sends a current time and date to the implantable device for the implantable device to keep on record. Subsequently, when the processor 302 is instructed to send altered parameters to the implantable device, the processor 302 performs steps 406, 407

and then in step 1002 transmits the stored time and date to the implantable device before proceeding with the remaining steps 408-410.

The implantable device's processor 204 performs steps 411-415 of the inquiry operation and in step 1003 receives and stores the date and time transmitted by the programming device. In the subsequent programming operation the processor 204 performs steps 416, 417 before receiving the transmitted time and date from the programming device in step 1004. Then in step 1006 the processor 204 compares the transmitted time and date to the stored time and date at which the last programming device successfully inquired or programmed the implantable device. If the numbers do not match, the processor 204 transmits a denial to the programming device in step 1008. Otherwise the processor 204 proceeds to receive the new parameters in step 418. Upon successful reception of all the transmitted parameters, the processor 204 sends a confirm signal in step 419 and stores the new parameters.

Referring now to Figure 11, the inquiry operations of both the programming device and implantable device are the same as those of Fig. 10. In step 1105, the implantable device's processor 204 sends the stored date and time after establishing the implantable device's identity in steps 416-417, and before performing the remaining programming steps 418-420. After steps 406-407 of the programming operation, the programming device's processor 302 receives the date and time of last successful inquiry operation from the implantable device in step 1101. In step 1103, the processor 302 compares the received date and time to its own stored date and time of last successful operation to determine if any other programming device has successfully interacted with the implantable device since the current programming device's last inquiry operation. If the access date and times are the same, the processor 302 proceeds with steps 408-410 of the programming operation. Otherwise, the processor 302 terminates the program operation in step 410 and preferably notifies the user of the error.

A preferred system embodiment is shown in Figure 12. In this embodiment, a time and date is associated with the last successful programming of the implantable device 106. After the programming device's processor 302 performs steps 401-405 of the inquiry operation, in step 1201 it receives and stores the time and date of the last programming operation on the implantable device. In any subsequent programming

operation, the processor 302 establishes the identity of the implantable device in steps 406-407 and sends the stored time and date in step 1202 before sending any parameters in step 408. After transmitting the parameters, the processor 302 sends the current date and time to the implantable device. The processor 302 then completes the programming operation steps 409-410.

After the implantable device's processor 204 performs inquiry operation steps 411-415, it sends the time and date of the last programming operation in step 1204. In the subsequent programming operation, the processor 204 establishes the implant's identity to the programming device in steps 416-417. In step 1205, the processor 204 receives the programming device's record of the last programming date and time, and compares it with the implantable device's record of the last programming date and time in step 1206. If they do not match, then the implantable device has been reprogrammed since the last inquiry operation by the programming device, and the processor 204 transmits a denial in step 1208 and terminates the programming operation in step 420. Otherwise, the processor 204 accepts the new parameters in step 418, and the current date and time in step 1210. If the programming operation is successful, the implantable device stores the current date and time as the new programming date and time, and in step 419 sends a confirmation to the programming device.

A more preferred embodiment is shown in Figure 13. The inquiry operation is the same as the inquiry operation of the embodiment shown in Figure 12. The programming operation differs for the implantable device in that after transmitting its identification information in step 417, the processor 204 transmits the stored programming date and time to the programming device in step 1305. The processor 204 then accepts any transmitted modified parameters in step 418 and if the transmission is successful, accepts and stores the current date and time in step 1210 before sending a confirmation in step 419.

The programming operation for the programming device begins with the processor 302 requesting and receiving the implantable device's identification information in steps 406-407. In step 1301 the processor receives the transmitted date and time from the implantable device and in step 1303 compares the transmitted date and time to the stored programming date and time. If they match, then none of the

configuration parameters has been re-programmed since the programming device's last inquiry operation. Consequently, the processor 302 proceeds to send the new parameters in step 408 and the current date and time in step 1203. Otherwise, the programming device ends the program operation in step 410 and preferably notifies the operator of the error.

The embodiments of Figs. 12 and 13 are preferred relative to the previously described embodiments since in these embodiments, multiple programming devices can simultaneously be qualified to provide a new set of parameters to the implantable device. Any programming device with a current version of the configuration parameters (as indicated by the associated programming date and time) can successfully program the implantable device. Conversely, in the other previously described embodiments, only a single programming device at a time (the one which has most recently interacted with the implantable device) can successfully program the implantable device.

Referring now to Figure 14, yet another embodiment is shown. In this embodiment, prior to each programming operation, programming device 110 inquires the configuration parameters in order to verify the accuracy of programming device 110's copy of those parameters. The inquiry operations for the implantable device and the programming device is the same as that of Figure 4. For the programming operation, the implantable device's processor 204 performs steps 416-417 to establish its identity to the programming device. Then in step 1406 the processor 204 receives a parameter request similar to that of step 414, and in step 1407 responds to the parameter request by sending the requested parameters. Then in step 418 the processor 204 may receive modified parameters from the programming device, and in step 419 the processor 204 acknowledges a successful receipt of the parameters and updates the current parameters with the new parameter values.

The programming device's processor 302 performs steps 406-407 to establish the implantable device's identity, then in step 1401, the processor 302 transmits a request for the configuration parameters similar to that of step 404. The processor 302 receives the transmitted parameters in step 1403 and in step 1405 compares them with the parameters from the previous inquiry. If there is no change (or only a slight change which may be attributed to adaptation or progression of a programmed algorithm), then the implantable

device has not been reprogrammed since the inquiry. The processor 302 consequently transmits the new configuration parameters in step 408. Otherwise, the programming device terminates the programming operation in step 410 and preferably notifies the user of the failure.

- 5 Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

WHAT IS CLAIMED IS:

1. A method for programming an implantable device, wherein the method comprises:
 - verifying that a programming device is aware of an implantable device's current parameters; and
 - communicating updated parameters from the programming device to the
5 implantable device.
2. The method of claim 1, wherein the verifying includes:
 - the programming device transmitting a programming device identification
 number to the implantable device; and
 - the implantable device comparing the programming device identification number
5 to a stored identification number which identifies a programming device
 that has made a most-recent inquiry of the implantable device.
3. The method of claim 2, further comprising:
 - sending a status message from the implantable device to the programming device,
 wherein the status message is a confirmation if the verification is
 successful and the updated parameters are successfully received, and
5 wherein the status message is a denial if the verification is unsuccessful.
4. The method of claim 1, wherein the verifying includes:
 - the implantable device transmitting a stored identification number to the
 programming device, wherein the stored identification number identifies
 a programming device that has made a most recent inquiry of the
5 implantable device; and
 - the programming device comparing the stored identification number to the
 programming device's own programming device identification number.
5. The method of claim 4, further comprising:
 - aborting the communication of updated parameters from the programming device
 to the implantable device if the verification is unsuccessful.

6. The method of claim 1, wherein the verifying includes:
the programming device determining if a predetermined time interval has elapsed
since the programming device's most recent inquiry.

7. The method of claim 6, further comprising:
aborting the communication of updated parameters from the programming device
to the implantable device if the predetermined time interval has elapsed.

8. The method of claim 6, wherein the predetermined time interval is within an inclusive
range between 5 minutes and 60 minutes.

9. The method of claim 6, wherein the predetermined time interval is within an inclusive
range between 10 minutes and 20 minutes.

10. The method of claim 1, wherein the verifying includes:
the implantable device determining if a predetermined time interval has elapsed
since a most recent inquiry of the implantable device.

11. The method of claim 10, further comprising:
sending a status message from the implantable device to the programming device,
wherein the status message is a confirmation if the predetermined time
interval has not elapsed and the updated parameters are successfully
received, and wherein the status message is a denial if the predetermined
time interval has elapsed.

12. The method of claim 10, wherein the predetermined time interval is within an
inclusive range between 5 minutes and 60 minutes.

13. The method of claim 10, wherein the predetermined time interval is within an
inclusive range between 10 minutes and 20 minutes.

14. The method of claim 1, wherein the verifying includes:
the programming device transmitting an inquiry time to the implantable device;
and
the implantable device comparing the inquiry time to a stored time which
indicates a time at which a most-recent inquiry of the implantable device
was made.
15. The method of claim 14, further comprising:
sending a status message from the implantable device to the programming device,
wherein the status message is a confirmation if the verification is
successful and the updated parameters are successfully received, and
wherein the status message is a denial if the verification is unsuccessful.
16. The method of claim 1, wherein the verifying includes:
the implantable device transmitting a stored time to the programming device,
wherein the stored time indicates a time at which a most-recent inquiry of
the implantable device was made; and
the programming device comparing the stored time to the programming device's
last inquiry time.
17. The method of claim 16, further comprising:
aborting the communication of updated parameters from the programming device
to the implantable device if the verification is unsuccessful.
18. The method of claim 1, wherein the verifying includes:
the programming device transmitting a stored time to the implantable device,
wherein the stored time indicates a time at which a most-recent
programming of the implantable device was performed; and
the implantable device comparing the stored time to the implantable device's
most-recent programming time.

19. The method of claim 18, further comprising:
communicating a current time along with the updated parameters from the
programming device to the implantable device; and
storing the current time as the most-recent programming time if the verification
is successful and the updated parameters are successfully received.
20. The method of claim 18, further comprising:
sending a status message from the implantable device to the programming device,
wherein the status message is a confirmation if the verification is
successful and the updated parameters are successfully received, and
wherein the status message is a denial if the verification is unsuccessful.
21. The method of claim 1, wherein the verifying includes:
the implantable device transmitting a most-recent programming time to the
programming device; and
the programming device comparing the most-recent programming time to a
stored time which indicates a time at which a most-recent programming
of the implantable device was performed.
22. The method of claim 21, wherein the stored time is provided to the programming
device from the implantable device during an inquiry operation.
23. The method of claim 21, further comprising:
communicating a current time along with the updated parameters from the
programming device to the implantable device; and
the implantable device storing the current time as the most-recent programming
time if the verification is successful and the updated parameters are
successfully received.

24. The method of claim 21, further comprising:

aborting the communication of parameters from the programming device to the implantable device if the verification is unsuccessful.

25. The method of claim 1, wherein the verifying includes:

the implantable device transmitting current parameters to the programming device; and

the programming device comparing the current parameters to stored parameters
5 from the programming device's last inquiry of the implantable device.

26. The method of claim 25, further comprising:

aborting the communication of parameters from the programming device to the implantable device if the verification is unsuccessful.

27. A multiprogrammer system for monitoring and optimizing implantable device performance, wherein the system comprises:

an implantable device that includes:

a memory configured to store configuration parameters;

5 a telemetry module configured to transmit and receive external communications; and

a microprocessor coupled to the memory to read and write configuration parameters, and further coupled to the telemetry module to send and receive configuration parameters, wherein the microprocessor
10 is configured to perform an algorithm in a manner governed by the configuration parameters; and

at least two programming devices each configurable to communicate with the telemetry module to receive configuration parameters from the implantable device and to send updated configuration parameters to the
15 implantable device,

wherein each one of the programming devices is configured to verify that configuration parameters retrieved by said one of the programming

devices are current before said one of the programming devices sends updated configuration parameters to the implantable device.

28. The multiprogrammer system of claim 27, wherein said one of the programming devices verifies that the retrieved configuration parameters are current by retrieving a stored programming device identification number from the implantable device and comparing the stored programming device identification number to an identification number of said one of the programming devices, whereby said one of the programming devices determines whether said one of the programming devices has made a most-recent inquiry of the implantable device.

29. The multiprogrammer system of claim 27, wherein said one of the programming devices verifies that the retrieved configuration parameters are current by determining if a predetermined time interval has elapsed since a most recent inquiry by said one of the programming devices.

30. The multiprogrammer system of claim 27, wherein said one of the programming devices verifies that the retrieved configuration parameters are current by retrieving a stored time from the implantable device and comparing the stored time to a last inquiry time of said one of the programming devices, whereby said one of the programming devices determines whether said one of the programming devices has made a most-recent inquiry of the implantable device.

31. The multiprogrammer system of claim 27, wherein said one of the programming devices verifies that the retrieved configuration parameters are current by receiving a most-recent programming time from the implantable device and comparing the most-recent programming time to a stored programming time provided by the implantable device during a last inquiry by said one of the programming devices.

32. The multiprogrammer system of claim 27, wherein said one of the programming devices verifies that the retrieved configuration parameters are current by retrieving the

current configuration parameters from the implantable device and comparing the current configuration parameters to the previously retrieved configuration parameters.

33. A programming device for an implantable device, wherein the programming device comprises:

- a user interface;
- a memory configured to store configuration parameters;
- 5 a communications circuit configurable to communicate with an implantable device; and
- a microprocessor coupled to the memory to read and write configuration parameters, coupled to the user interface to display the configuration parameters, and further coupled to the communications circuit to retrieve configuration parameters from the implantable device and to send configuration parameters to the implantable device, wherein the microprocessor is configurable to retrieve, write, read, alter, and send the configuration parameters in response to input from the user interface,
- 10 wherein the microprocessor is configured to verify that the retrieved configuration parameters are current before sending altered configuration parameters to the implantable device.

34. The programming device of claim 33, wherein the microprocessor verifies that the retrieved configuration parameters are current by retrieving a stored programming device identification number from the implantable device and comparing the stored programming device identification number to an identification number of the implantable
- 5 device programming device, whereby the microprocessor determines whether the implantable device programming device has made a most-recent inquiry of the implantable device.

35. The programming device of claim 33, wherein the microprocessor verifies that the retrieved configuration parameters are current by determining if a predetermined time

interval has elapsed since a most recent inquiry of the implantable device by the microprocessor.

36. The programming device of claim 33, wherein the microprocessor verifies that the retrieved configuration parameters are current by retrieving a stored time from the implantable device and comparing the stored time to a last inquiry time of the programming device, whereby the microprocessor determines whether the programming device has made a most-recent inquiry of the implantable device.

37. The programming device of claim 33, wherein the microprocessor verifies that the retrieved configuration parameters are current by receiving a most-recent programming time from the implantable device and comparing the most-recent programming time to a stored programming time provided by the implantable device during a last inquiry by the microprocessor.

38. The programming device of claim 33, wherein the microprocessor verifies that the retrieved configuration parameters are current by retrieving the current configuration parameters from the implantable device and comparing the current configuration parameters to the previously retrieved configuration parameters.

39. A multiprogrammer-aware implantable device which comprises:

- a memory configured to store configuration parameters;
- a telemetry module configured to transmit and receive external communications;
- and

- a microprocessor coupled to the memory to read and write configuration parameters, and further coupled to the telemetry module to send configuration parameters to a programming device and to receive configuration parameters from the programming device, wherein the microprocessor is configured to perform an algorithm in a manner governed by the configuration parameters, and

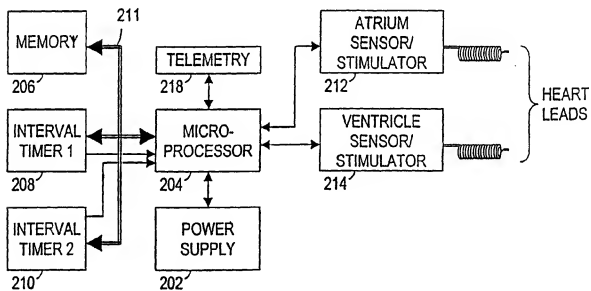
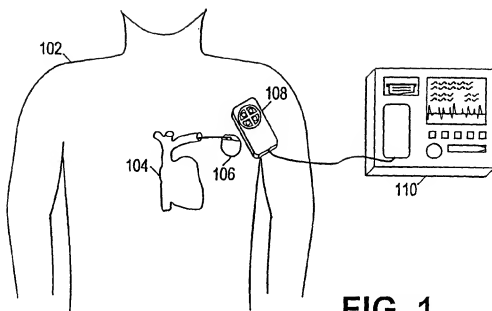
wherein the microprocessor is configured to verify that the programming device has received a current version of the configuration parameters before accepting altered configuration parameters.

40. The implantable device of claim 39, wherein the microprocessor verifies that the programming device has received a current version of the configuration parameters by receiving a programmer identification number from the programming device and compares the programmer identification number to a stored identification number which
5 identifies a programming device that has made a most-recent inquiry of the implantable device.

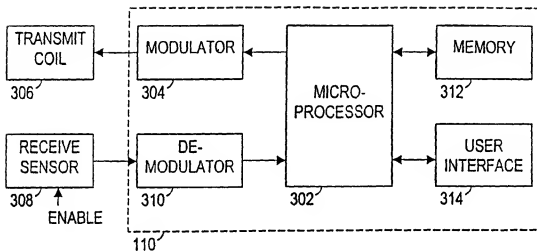
41. The implantable device of claim 39, wherein the microprocessor verifies that the programming device has received a current version of the configuration parameters by determining if a predetermined time interval has elapsed since a most recent inquiry of the implantable device.

42. The implantable device of claim 39, wherein the microprocessor verifies that the programming device has received a current version of the configuration parameters by receiving an inquiry time from the programming device and comparing the inquiry time to a stored time which indicates a time at which a most-recent inquiry of the implantable
5 device was made.

43. The implantable device of claim 39, wherein the microprocessor verifies that the programming device has received a current version of the configuration parameters by receiving a programming time from the programming device and comparing the programming time to a stored time which indicates a time at which a most-recent
5 programming of the implantable device was performed.



106 ↗

**FIG. 3**

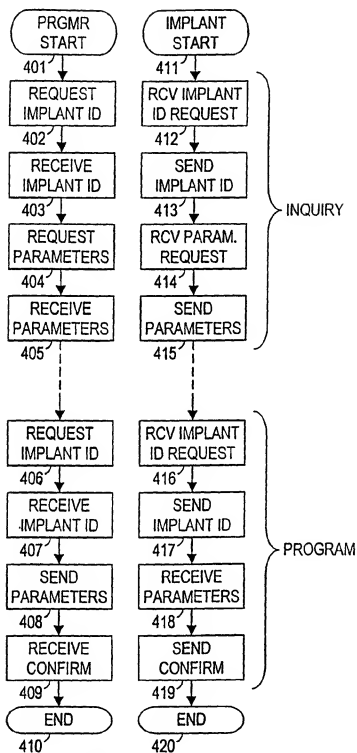


FIG. 4

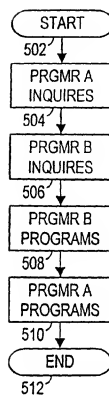


FIG. 5

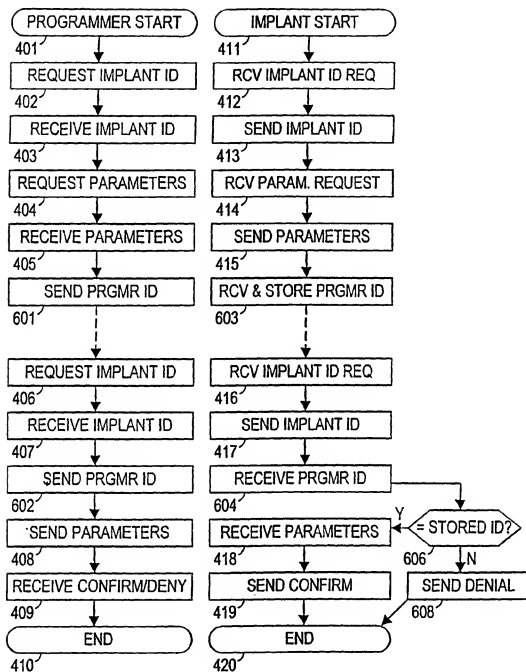


FIG. 6

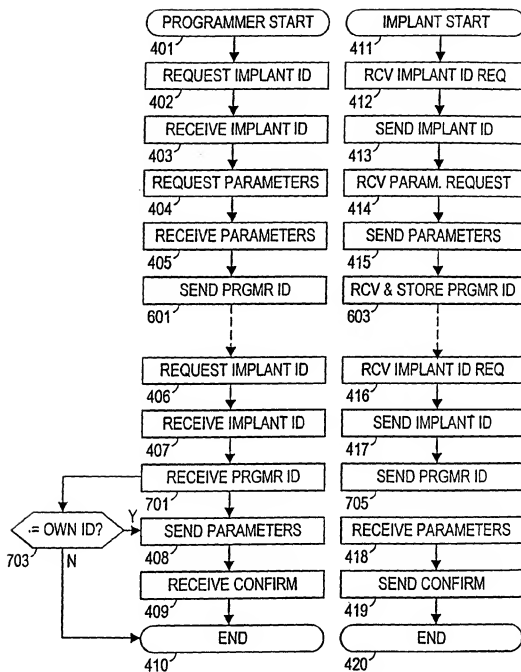


FIG. 7

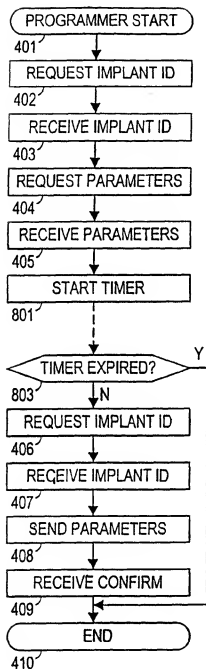


FIG. 8

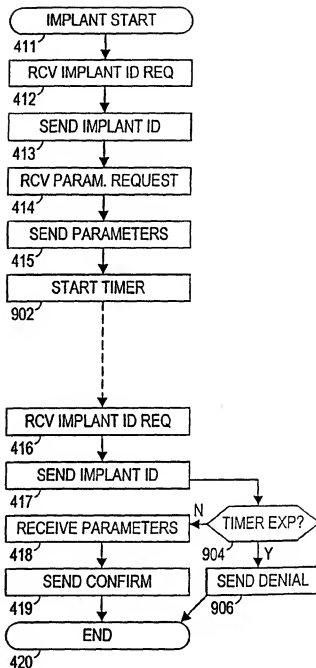


FIG. 9

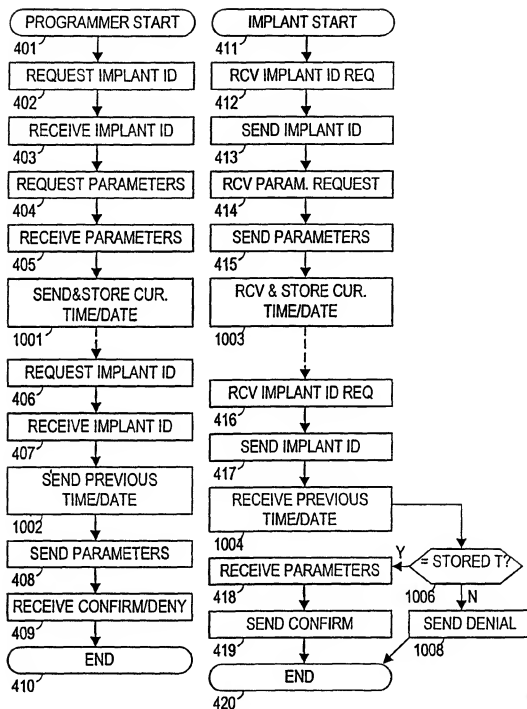


FIG. 10

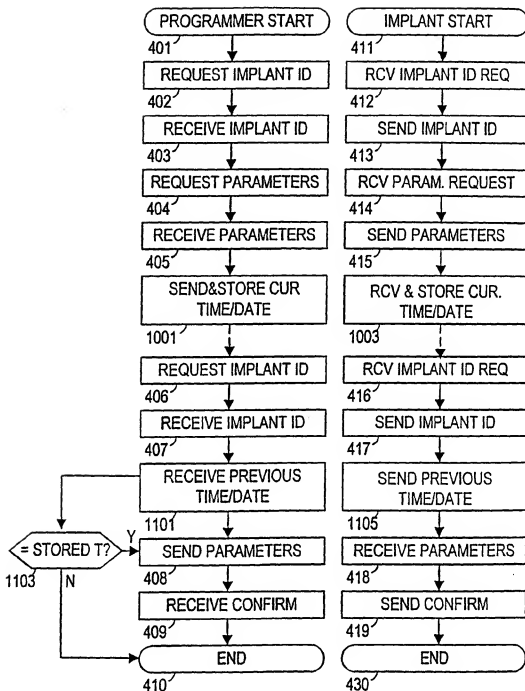


FIG. 11

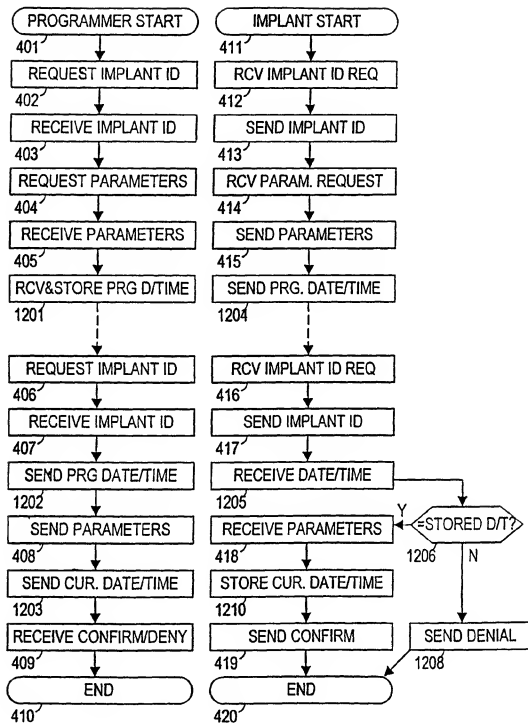


FIG. 12

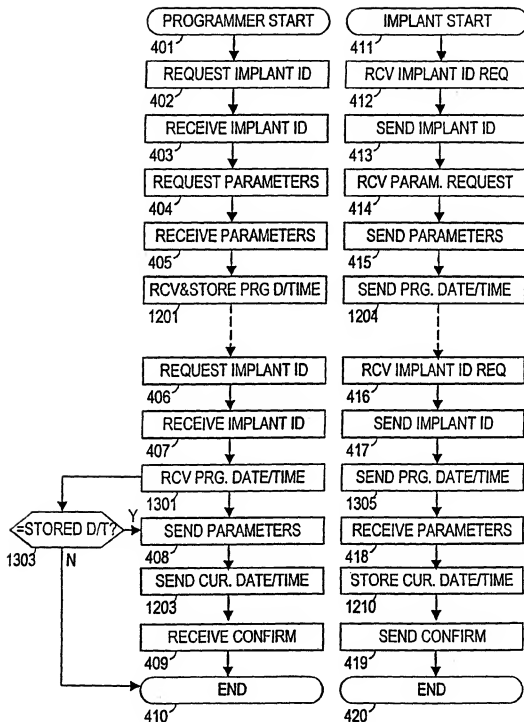


FIG. 13

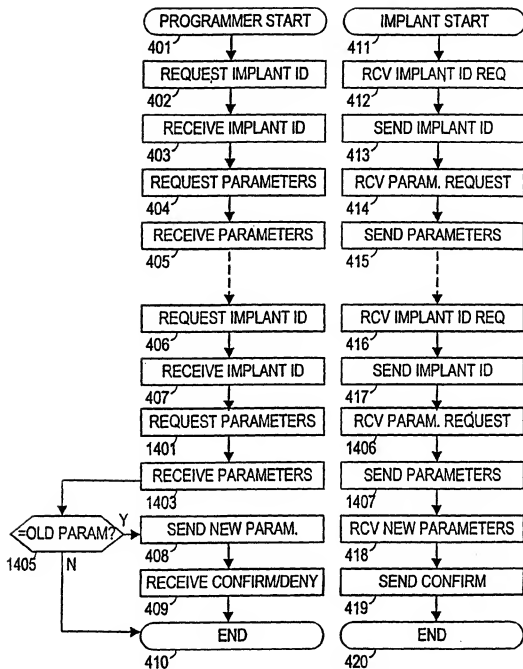


FIG. 14

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/US 99/27056

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 A61N1/372

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 A61N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 725 559 A (ALT ECKHARD ET AL) 10 March 1998 (1998-03-10) column 8, line 9-46	1,27,33, 39
A	US 5 690 690 A (CHINN STEVE ET AL) 25 November 1997 (1997-11-25) column 2, line 21-67 column 7, line 49-56	1,27,33, 39
A	US 4 365 633 A (LOUGHMAN JAMES ET AL) 28 December 1982 (1982-12-28) column 1, line 36 -column 2, line 40	1,27,33, 39

☐ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"B" document member of the same patent family

Date of the actual completion of the international search

23 March 2000

Date of mailing of the international search report

30/03/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentisten 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tlx. 31 651 epo nl,
Fax (+31-70) 340-3016

Authorized officer

Grossmann, C.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/27056

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5725559 A	10-03-1998	CA 2255653 A	20-11-1997
		EP 0906137 A	07-04-1999
		WO 9743004 A	20-11-1997
US 5690690 A	25-11-1997	EP 0730882 A	11-09-1996
US 4365633 A	28-12-1982	CA 1154832 A	04-10-1983
		DE 3043999 A	10-09-1981
		FR 2481933 A	13-11-1981
		GB 2069844 A,B	03-09-1981
		JP 1606406 C	13-06-1991
		JP 2033391 B	26-07-1990
		JP 56156163 A	02-12-1981
		NL 8006373 A	16-09-1981
		SE 449443 B	04-05-1987
		SE 8100624 A	23-08-1981